LISTING OF CLAIMS

1. (currently amended) A method for distributed computation of an RSA inverse value (y) for asynchronous authentication for communication of encrypted messages in an asynchronous network for at least two input values (x, e) among n-1 participating network devices compromising $t < n/4$ faulty devices and a non-faulty leader device, the participating network devices holding share values of the Euler function (μ(N)) of RSA modulus (N), each participating network device performing the steps of:
    a. choosing a first random value (q) and a second random value (r);
    b. sharing over integers (Z) the first random value (q), the second random value (r), and the zero value (0);
    c. the leader device performing additionally the steps of:
        i. receiving a first, second, and third sub-share value (q_i, r_i, 0_i) from at least $t + 1$ participating network devices;
        ii. broadcasting ~~the~~ identities of said participating network devices;
    d. receiving the identities and corresponding sub-share values (q_i, r_i, 0_i);
    e. deriving a sum-share value (F) from one of the <u>sub-share</u> ~~share~~ values, the at least one input value (e), and the corresponding first, second and third sub-share values (q_i, r_i, 0_i) defined by the identities<u>;</u>
    f. broadcasting the sum-share values (F)<u>;</u>
    g. receiving $2t + 1$ sum-share values (F_i)<u>;</u>
    h. deriving a polynomial (f) interpolating the sum-share values (F_i) and an exponent share value (d_P) dependent on the polynomial (f), and an inverse-

share value (y_P) dependent on the exponent share value (d_P) and the RSA modulus (N);

i. broadcasting the inverse-share value (y_P);

j. receiving $t + 1$ inverse share values ~~value~~ (y_i);

k. verifying the validity of the received values for authenticating participating network devices by obtaining the RSA inverse value (y) from the received inverse-share values ~~value~~ (y_i), determining the validity of the obtained RSA inverse value (y) using at least two input values (x, e,), and, in the event of a positive determination, broadcasting the RSA inverse value (y), and stopping further calculations ; and

l. applying the derived exponent share values for encryption of communications with authenticated participating network devices .

2. (original) A method according to claim 1 to compute the RSA inverse value (y), wherein among n participating network devices (A, B, C, D) compromising $t < n/4$ faulty devise at least $t + 1$ participating network devices (A, B, C, D) act as a leader device while performing n times the steps of claim 1.

3. (canceled)

4. (currently amended) A method according to claim 1, wherein the step (b) ~~(II)~~ of sharing over integers (Z) comprises using a threshold signature for determining the consistency of subsequently received sub-share values (q_i, r_i, 0_i).

5. (currently amended) A method according to claim 1, wherein step (b) ~~(II)~~ of sharing over integers (Z) comprises using a vector of digital signatures for determining the consistency of subsequently received sub-share values (q_i, r_i, 0_i).

6. (currently amended) A method according to claim 1, wherein step (h) ~~(VIII)~~ of deriving an exponent share value (d_P) comprises using the Extended Euclidean Algorithm.

7. (currently amended) A method according to claim 1, wherein the step (k) ~~(XI)~~ of obtaining the RSA inverse value (y) from the received inverse-share value (y_i) comprises using the Lagrange Interpolation Algorithm.

8. (currently amended) A plurality of program storage devices, each readable by at least one digital processing apparatus and having a program of instructions which are tangibly embodied in the storage devices and which are executable by at least one processing apparatus to perform a method for distributed computations of an RSA inverse value (y), for asynchronous authentication for communication of encrypted messages in an asynchronous network from at least two input values (x, e) among $n - 1$ participating network devices (A, B, C, D) holding share values (μ_A, μ_B, μ_C, μ_D) of the Euler function ($\mu(N)$) of an RSA modulus (N), said program allowing each participating network to perform the following steps:

    a. choosing a first random value (q) and a second random value (r);

b. sharing over integers (Z) the first random value (q), the second random value (r), the zero value (0);

c. the leader device (D) performing additionally the steps of:

   i. receiving a first, second, and third sub-share value (q_i, r_i, 0_i) from at least $t + 1$ participating network devices;

   ii. broadcasting ~~the~~ identities (S) of said participating network devices

d. receiving the identities (S) and corresponding sub-share values (q_i, r_i, 0_i);

e. deriving a sum-share value (F) from the sub-share ~~share~~ value ~~(μ_P)~~, the at least one input value (e), and the corresponding sub-share values (q_i, r_i, 0_i) defined by the identities (S),

f. broadcasting the sum-share value (F);

g. receiving 2t + 1 sum-share values (F_i);

h. deriving a polynomial (f) interpolating the sum-share ~~sum0share~~ values (F_i) and an exponent share value (d_P) dependent on the polynomial (f) and an inverse-share value (y_P) dependent on the exponent share value (d_P) and the RSA modulus (N)[[.]] ;

i. broadcasting the inverse-share value (y_P);

j. receiving t + 1 inverse-share values ~~value~~ (y_i);

k. verifying the validity of the received values for authenticating participating network devices by obtaining the RSA inverse value (y) from the received inverse-share values ~~value~~ (y_i), determining the validity of the obtained RSA inverse value (y) using at least two input values (x, e,), and,

in the event of a positive determination, broadcasting the RSA inverse value (y), and stopping further calculations ; and

l. applying the derived exponent share values for encryption of communications with authenticated participating network devices .